Available online at www.sciencedirect.com
SCIENCE DIRECT®
The Journal of Systems and Software 79 (2006) 120–131
The Journal of Systems and Software
www.elsevier.com/locate/jss

A method for defining IEEE Std 1471 viewpoints

Henk Koning *, Hans van Vliet

Faculty of Science, Vrije Universiteit, Amsterdam, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

Received 7 December 2004; received in revised form 7 February 2005; accepted 11 February 2005
Available online 8 April 2005

Abstract

With the growing impact of information technology the proper understanding of IT-architecture designs is becoming ever more important. Much debate has been going on about how to describe them. In 2000, the IEEE Std 1471 proposed a model of an architecture description and its context.

In this paper we propose a lightweight method for modeling architectural information after (part of) the conceptual model of IEEE Std 1471 and defining IEEE Std 1471 viewpoints. The method gives support by outlining in textual form and in diagram form the relation of the concerns of the stakeholders to the architectural information. The definition of viewpoints can then be done with insight from these relations. The method has four steps: (1) creating stakeholder profiles, (2) summarizing internal design documentation, (3) relating the summary to the concerns of the stakeholders, and (4) defining viewpoints.

We have conducted a round of discussion and testing in practice in various settings. In this paper we present the feedback we received and propose improvements.

Keywords: Architecture; IEEE Std 1471; Viewpoints

1. Introduction

IT architecture is a relatively new branch within software engineering. IEEE Std 1471 (IEEE, 2000) defines it as "Architecture is the fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution". van Vliet (2000) places the architecture definition phase in the software life cycle between the requirements engineering and design phases. In this phase the interests and concerns of all stakeholders are taken into account to come to a well-balanced solution.

Current practice is that designers of IT architectures are predominantly problem-driven. Design often is a fuzzy and non-rational process, see Parnas and Clements (1986), but after arriving at a balanced solution which solves the problem, the architect describes the solution in a structured way. This can be a one-time structuring or a structure following a known framework such as those of Kruchten (1995); Soni et al. (1995), or Boar (1998). Clements et al. (2003) offers many helpful models and guidelines for composing an architecture description. Using a one document framework for all stakeholders can mean for a certain stakeholder that the information that is relevant to his concerns can be very scattered, see Koning and van Vliet (submitted for publication).

In 2000, the IEEE Std 1471 proposed a model of an architecture description and its context. It offers a high level generic model for architecture descriptions with explicit attention to the concerns of the stakeholders.

* Corresponding author. Tel.: +31 6 2042 4278; fax: +31 20 598 7728.
E-mail addresses: henk@cs.vu.nl (H. Koning), hans@cs.vu.nl (H. van Vliet).

0164-1212/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2005.02.023

In this paper we offer support for the application of this model.

The organization of this paper is as follows: in Section 2 we lay out the research setting. In Section 3 we give a description of the method and show examples of the deliverables of each step. In Section 4 we outline the validation activities and present the results for each step. In Section 5 we draw our conclusions and propose improvements. Section 6 summarizes future work.

2. Research setting

In this section we lay out the research setting. We first introduce IEEE Std 1471 and state our position towards this standard. We then describe our project approach. We close this section with listing some assumptions on which our method is based.

2.1. IEEE Std 1471

IEEE Std 1471 describes a model of an architecture description (AD) and its context (IEEE, 2000). On page 1, it says: "The purpose of this recommended practice is to facilitate the expression and communication of architectures". On page 2: "Furthermore, it establishes a conceptual framework of concepts and terms of reference within which future developments in system architectural technology can be deployed. This recommended practice codifies those elements on which there is consensus; specifically, the use of multiple views, reusable specifications for models within views, and the relation of architecture to system context."

Central 'terms of reference' in the IEEE 1471 conceptual model are 'views', 'viewpoints', 'stakeholders' and 'concerns'. An 'architectural description' consists of 'views' that are each made according to a 'viewpoint' (see Fig. 1). According to the conceptual model a stakeholder is represented by his concerns.

A view is "A representation of a whole system from the perspective of a related set of concerns" (id, p. 9), and a viewpoint is "A specification of the conventions for constructing and using a view. A pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis." (id, p. 10). Viewpoints delineate the architectural information that is presented to the stakeholder. A viewpoint on the one hand prescribes the content and 'models' to be used, and, on the other hand, it indicates its intended 'stakeholders' and their 'concerns'.

The standard lists a number of essential stakeholders and concerns, and gives examples of the use of architecture description and of some viewpoints. The standard gives no general guidance for defining viewpoints. It only states that a viewpoint addresses a set of related concerns and that the viewpoints together should cover all the concerns of the stakeholders. There are no criteria given to decide on the 'relatedness' of concerns.

With respect to our interest in communication of architecture, the main contribution of IEEE 1471 is the explicit orientation on stakeholders and concerns. Following the path from his recognized concerns via the prescriptions in the viewpoint a stakeholder should be able to find the information of his interest in the views.

We also believe there are some drawbacks to this standard, see Koning and van Vliet (submitted for publication). We feel IEEE Std 1471 should be extended with guidance on how to achieve document qualities like "accessibility" to and "understandability" for the stakeholders.

Our research focuses on the definition of viewpoints as the leverage point to improve the quality of the architecture description, and more particular on improving the insight in the relation of the architecture design to the concerns of the stakeholders, before deciding on the viewpoints to use.

Application of this method contributes to meeting clauses 5.2 and 5.3 of IEEE 1471. It can also be used to construct library viewpoints or evaluate existing library viewpoints for possible use in a given situation.

2.2. Project approach

This research project follows an "action research" approach, see Baskerville (1999). In action research five steps are defined: diagnosing, action planning, action taking, evaluation, specifying learning. Our diagnosing of four real life IT-architecture documents has raised serious doubts about whether the stakeholders could find the information they needed. Our action planning resulted in the method for designing IEEE 1471 viewpoints described in this paper. Our action taking has been discussion and small scale testing.

The action research participants were IT architects of two companies, ING and Ordina, and students from our faculty. ING is a Dutch international bank that attaches great importance to IT architecture to manage its very complex IT operations. Over 10,000 people are working in their IT departments worldwide, among them several hundred IT architects. Ordina is an IT-consulting firm. It has been developing a view on managing and documenting large IT processing environments for some years. Ordina Public Consulting has government and municipal organizations as customers. The students were taking part in a course on software architecture.

We were actively involved in the discussions and test sessions described in this paper as presenters and moderators. Notes were taken during all the sessions and shared with the persons present on the spot.

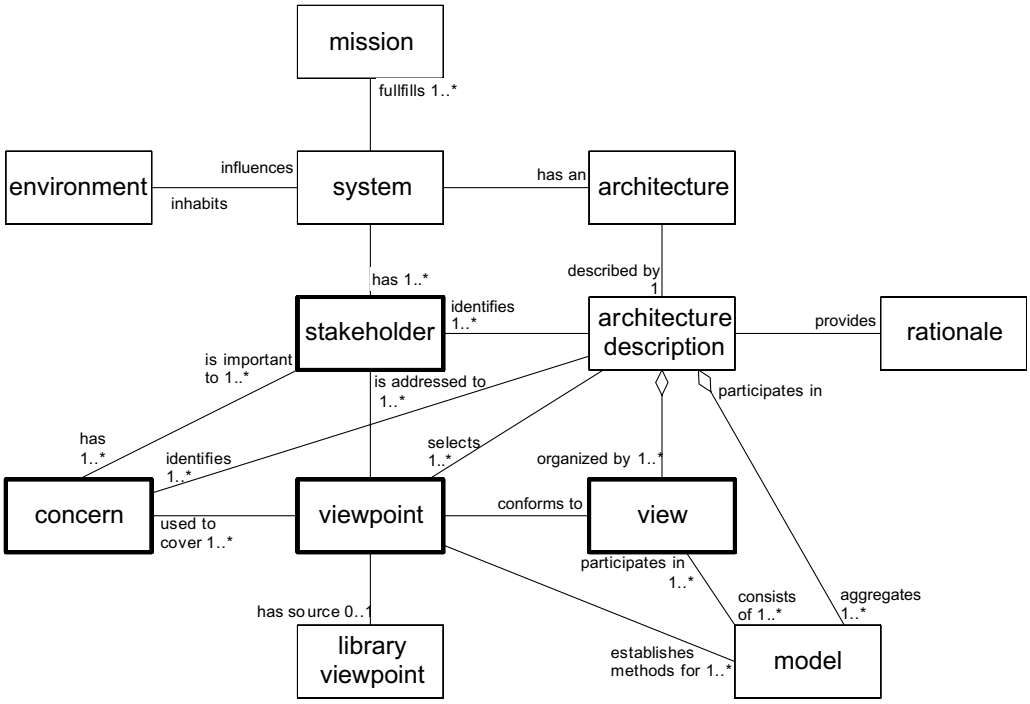

Fig. 1. The conceptual model of the IEEE 1471 standard for architectural descriptions (essential concepts for this study are indicated in bold; where no cardinality is indicated it is '1').

2.3. Assumptions

The method described in this document is based upon the following assumptions:

Designing an architecture is a fuzzy and non-rational process. For communication purposes the resulting design data need to be structured (Parnas and Clements, 1986).

Documenting the architecture is an activity that takes place in all stages of an architecture design project. For internal discussion or for intermediate discussion with stakeholders, parts of the problem statement and of the designed solution under consideration are described, altered, and described again. These pieces of description may have a varying degree of formality. There is a gradual shift from problem orientation to solution orientation.

Describing architecture at the moment requires a situational approach, which means, it is dependent on the peculiarities of the project at hand. Although many attempts have been made to standardize architecture descriptions, that is, to prescribe a fixed set of views, the current practice is that architects, for good reasons, make their own choices for each project.

Though the situational approach is common, it is not per se desirable. Where repeated use can be made of the same viewpoints, IEEE 1471 offers the possibility of storing and reusing viewpoints as library viewpoints.

An architecture description that is composed in a stakeholder oriented way is better readable for the stakeholder. Better readable means: the stakeholder can find more quickly the information that is relevant for him, and he can process that particular information more easily. Stakeholder orientation is determined by the structure of the document (the division in views and the outline within each view should be relevant to him), and the use of text and diagrams (words and graphics should be meaningful to him).

The smaller the number of views a stakeholder must consult to see how his concerns are addressed, the better it is. The smaller the amount of unnecessary information a view contains for a stakeholder, the better it is. Information that is only for internal use by the architecture team should not be communicated to the stakeholders.

Understanding and evaluating an IT architecture by a stakeholder is basically a process of translating the IT architecture concepts to his/her own concepts, and making inferences about possible situations or results that may occur by introducing the IT architecture.

Diagrams play an important role in the communication of IT architecture. Diagrams contribute mainly by giving an overview of components and relations and by giving support in making inferences, see Gyselinck and Tardieu (1998). Diagrams speed up the processing of the information and they aid in remembering the information.

Effective communication needs to be designed. The basic question is: what do you want to tell to whom?

Explicit representation (in text and graphics) of one's thoughts gives a better insight, leads to corrective thoughts and to a more complete design.

3. Viewpoint design method

3.1. Introduction

Our method for designing viewpoints consists of four activities that are, in principle, performed near the end of the design phase, before official decision taking and communication takes place. It embodies a roundup of what is already thought, said or written and puts it in a structure. If necessary, information can be added to get a more complete description. A partial application of this method earlier in the design project is feasible, and we actually expect that to happen in practice, but for testing purposes we do not want to make things too complicated at the moment.

Note: we like to mention here right away that the description 'near the end of the design phase' received some very straight criticism from the practitioners, even with the added nuances. Their comment: stakeholders and communication are essential throughout the whole design project. See further our 'conclusion' section.

An important aspect to consider is the level of detail with which these activities are performed. This can be restricted to what is necessary for properly clustering the architectural information in a stakeholder oriented way. The descriptive texts used in the method can be very terse, only to be understood by the designer himself. Our advice is to perform these activities with some speed in an intuitive manner, see what picture emerges and, if felt necessary, iterate to add details or improvements.

The activities to perform are:

- Step 1: Compile stakeholder profiles.
- Step 2: Summarize internal design documentation.
- Step 3: Relate summary of internal design documentation to concerns of stakeholders.
- Step 4: Define viewpoints.

In Fig. 2 these activities have been positioned in relation to the essential concepts of IEEE Std 1471. Steps 1–3 together create a 'document design view' of the architecture description to be made, which is then translated into IEEE Std 1471 viewpoints.

Basically, these activities provide a means to tinker for some period of time with the relation between the essential content of the architecture design and the concerns of the stakeholders. Various tools are provided to express and amend the thoughts of the architect, find omissions, seek words to express the perceived relation, etc.

We summarize in the next sections the method as it was presented after some time in the test period. The original description of the method, see Koning (2003), contains an extra step 5 'test result with stakeholders', and has more details and more open advice and research questions. While presenting the method to the testers it was immediately clear that we had to restrict ourselves and we focused on the parts of the method that gave support for the clustering of architectural information in views.

The examples in the method description are from the ING target architecture project. The problem to be addressed there was that for a new financial reporting item, called 'market risk', the data is manipulated in many systems which leads to reporting errors and high maintenance costs. An extra viewpoint example is from a student assignment.

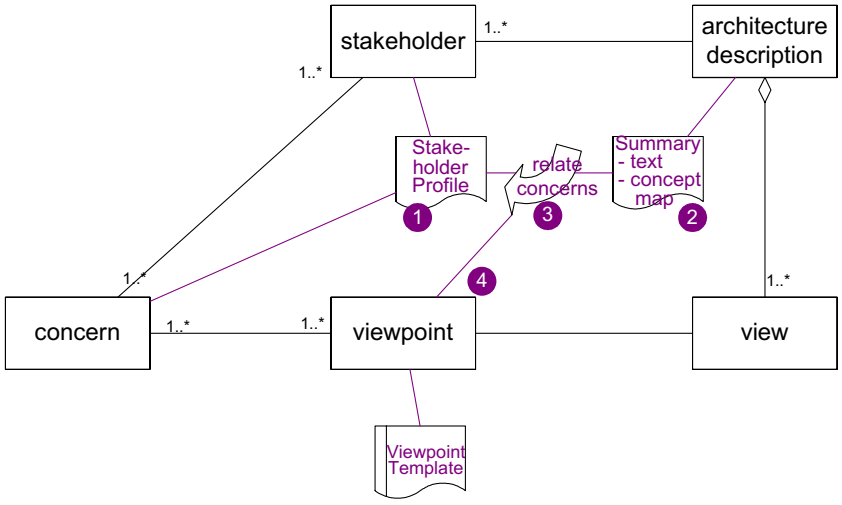

Fig. 2. Method viewpoints design, positioned to essential concepts of IEEE Std 1471.

3.2. Step 1: Compile stakeholder profiles

In this step a 'stakeholder profile' is compiled for each stakeholder that is relevant for this architecture design. A stakeholder profile is a simple table that holds descriptive text for five attributes: title, goals, tasks, concepts, concerns.

The goal of this activity is to make the stakeholder position explicit and to be able to reason about his information needs. The table expresses how the architect sees the stakeholder. It is a condensed résumé of the position of that stakeholder with respect to the problem domain, see Table 1. Uncovering new information is not the goal, but this activity may reveal that information is missing. If so, action can be undertaken to supply this information.

The choice of these attributes was inspired by literature from requirements engineering (Kotonya and Sommerville, 1998), and user interface design (van Welie, 2001). See Table 2 for an example of a stakeholder profile.

As far as the number of stakeholder profiles to produce is concerned, we consider five a good average and 10 a lot.

For an even better understanding two profiles can be made for each stakeholder. One general profile expressing the position of the stakeholder in the organization, and one expressing the position towards the architecture.

IEEE 1471, clause 5.2 mentions some stakeholders and concerns that should be taken into account as a minimum.

The stakeholder profiles can be adjusted and further developed when used in the next steps.

Table 1
Attributes of a stakeholder profile

Attribute	Meaning
Title	Short recognizable description of role/function
Goals	The goal or goals of the role/function. A goal is a condition that must be reached or maintained
Tasks	Logical grouping of all the activities that must be performed for the role/function
Concepts	Objects that are relevant for the stakeholder and that make up his view of the world
Concerns	Concrete interests or worries that guide the activities in the role/function and that determine which services are requested of other roles/functions

Table 2
Example of a stakeholder profile

Attribute	Content
Title	CEO finance
Goals	Deliver good financial planning and reporting
Tasks	• Financial reporting • Mgt reporting • Product control (profit&loss) • Raroc reporting
Concepts	Financial data, various kinds of reports, risks, budgets (medium term planning)
Concerns	• How do I deliver correct and reliable information? • How to receive/deliver financial data from/to other parties that are consistent with their reporting (Market Risks, Credit Risks)

3.3. Step 2: Summarize internal design documentation

In this step the available, internal design documentation is summarized (if not done already). With internal design documentation we mean any recording of information that circulates within the design team and is part of the current design as far as it has progressed. This information can be structured according to some formal prescription or not, that is up to the designer.

The goal is to produce an overview of the architectural information that makes it possible to reason about the relation of this information to the concerns of the stakeholders. The overview should name parts of the information that can be allocated to views. It must be sufficiently clear to aid memory and be able to reason about it.

This step has two deliverables: a textual summary of the internal design documentation, and a map of the key architectural concepts.

The textual summary is a short bulleted list of the main architectural decisions. See for Table 3 for an example.

A number of 5–10 statements seems reasonable.

It may seem strange to represent months of exploratory work and design deliberations in such a small list, but for communication it is necessary to create the 'top of the pyramid'. While you are in the process of designing it is not always evident what currently your short list of statements is. Compiling this textual summary is a good way to become aware of what your current short list is.

This list can be adjusted and further developed when used in the next steps.

The textual summary will probably contain a number of terms that are key architectural concepts.

Architectural key concepts are the 'things' that come to surface when you ask 'what kind of' questions, like

Table 3
Example of textual summary of internal design documentation

Main architectural statements for new application architecture of Market Risks
• One system for all regions and business units
• Use of a common data warehouse together with business unit Y
• Application functions built as services that can be invoked over a message bus
• Better systems
• Great reduction in number of internal and external interfaces

'what kind of components are part of the design?', or 'what kind of aspects did we look into?'. Examples of architectural concepts are: concerns, design principles, goals, applications, processes, products, infrastructure services, deadlines, money, design guidelines, etc. Concepts are part of the language of the architect. Concepts can be represented in a UML-class diagram. See Fig. 3 for an example map of key architectural concepts (with added concerns of step 3). It is a mixed bag of topics that are for whatever reason relevant to the design.

The map represents a meta-model of the description of the architecture design. It strongly shows how elements of the design are related. It is a thinking tool, not a database design, therefore the presentation stays informal.

3.4. Step 3: Relate summary of internal design documentation to concerns of stakeholders

In this step the results of step 1 and step 2 are related to each other. The goal of this activity is to make explicit the relation between the content of the architecture design and the concerns of the stakeholders. This relation is expressed in textual form and in graphical form. The texts reflect not only the fact that there is a relation, but also the essential reasoning that shows how the architectural statements address the concerns of the stakeholders.

The use in the next step (the design of the viewpoints, 'who needs to know what?') will determine the proper level of detail, which may not be easy to find right away. The text and graphics produced are of help in producing the final documentation and make that more smooth.

The textual expression of the relation of the content of the architecture design to the concerns of the stakeholders is expressed in a table (see Table 4 for an example). The first column contains the statements from the textual summary created in step 2. The table has two header rows: the second row contains the main concerns of the stakeholders. These concerns are derived from the stakeholder profiles created in step 1 by combining similar concerns. The first header row contains the stakeholders that share this concern.

At the crossing points of rows and columns, the question can be asked 'is this architectural statement (row) relevant for this concern (column)?'. If so, that cell will be filled with some descriptive text that explains what the architectural statement means for this concern. It may also reveal some of the details behind the architecture statement that are relevant for this concern. In the empty cell above some descriptive text will be put that makes the (general) concern more specific for this architectural statement.

The table is a thinking tool. It is not necessary to add text to all the crossing points. To arrive at a balanced

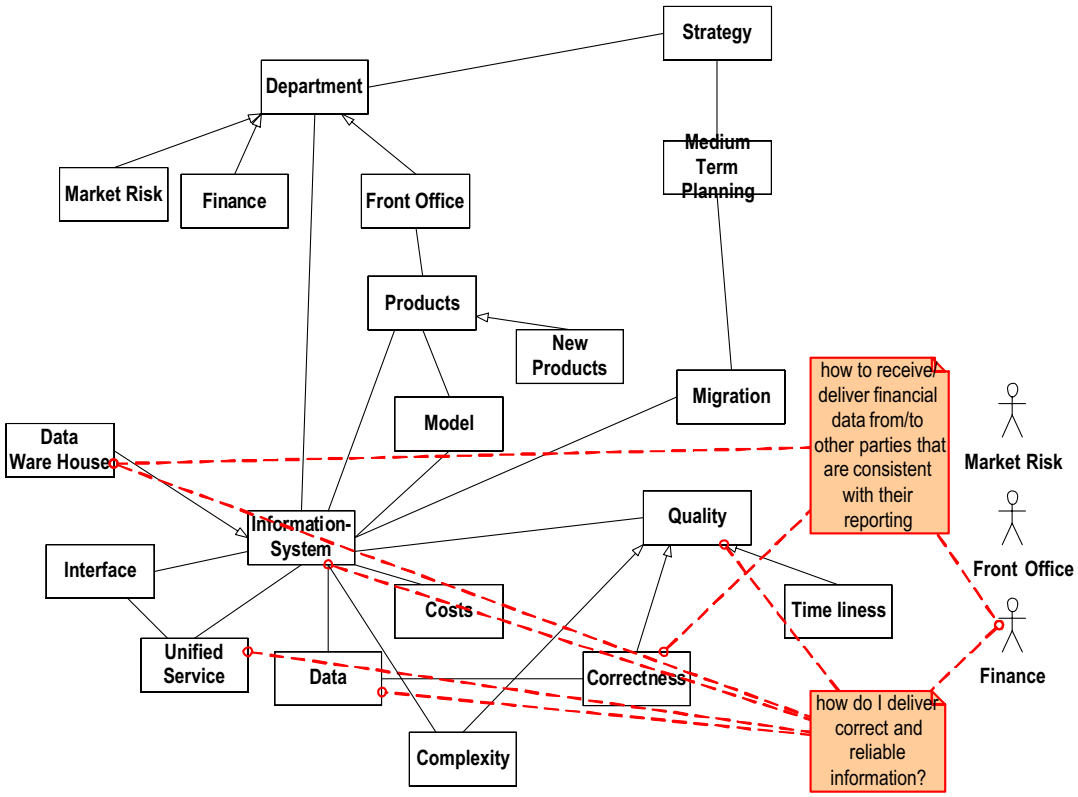

Fig. 3. Example of a concept map with concerns.

Table 4
Example of content-2-concerns table (see text for description of the structure of this information)

Architectural statements	Concerns			
	CEO, CIO	CEO, Business managers	Business managers	CIO
	The current situation is very fragmented (many errors, high costs, long maintenance cycles), how can this be straightened out?	*How can new products be introduced more quickly?*	*How can we improve the quality of our work?*	*How can I reduce system development costs?*
	How do I simplify the IT operations?	*How can the information systems be changed more quickly?*		
One system for all regions and business units	One system will greatly reduce the complexity, but the migration can be very complex	One system will bring time to market to 3 months (instead of 12)		
	How do I reduce system dependency?			
Use of a common data warehouse together with business unit Y	A common datawarehouse will reduce the number of interfaces from 50 to around 10			
				How can I reuse available components?
Application functions built as services that can be invoked over a message bus				Approx 70% of application functions can be reused by other systems over the message bus
			How can the systems support the work more?	
Better systems			Better models will be developed that support more exceptions to standard operations	

link between a statement and a concern some rephrasing may be needed of the statement or of the concern.

Making the architectural statements more specific is often more easy than making the concerns more specific. Making the concerns more specific is part of filling in the gap between the world of the stakeholder and the world of the architect, and it helps the architect in realizing what his statement means for the stakeholder.

The graphical expression of the relation of the content of the architecture design to the concerns of the stakeholders is based upon the concept map produced in step 2. This map is enriched with boxes that denote the stakeholders and their concerns. After that extra lines are drawn from the concerns to the architectural concepts in the map that are relevant to the concerns.

Fig. 3 shows a Concept Map with the concerns of one stakeholder, related to the architectural concepts that are relevant for these concerns.

The concept map can soon become very messy when the concerns are added and the lines are drawn. If the tooling permits, this can be relieved by putting concerns in layers that can be made invisible. Descriptive text can be added to the lines, this gives the diagram a more immediate meaning.

3.5. Step 4: Define viewpoints

The goal of this activity is to define IEEE 1471 viewpoints in a way that fully takes into account what the architecture design means to (the concerns of the) stakeholders. The outline of text, the use of terms and graphics is geared to the stakeholders.

The primary decision to be taken is how information shall be clustered in views. Next a viewpoint is defined for each view to be produced. It seems only natural to capture the reasons for the choices made in this step as part of the rationale.

During our testing activities in the past months we offered two templates to the testers: a short one and an extended one, see Table 5 (the extended attributes are in italics). These templates were produced by us during the testing in reaction to the demands of the testers. To compile the extended list of viewpoint attributes we made use of IEEE 1471 Clause 5.3, Clements et al. (2003, p. 317) and Hilliard (2001). We left the tester free-

Table 5
Experimental viewpoint template *with extended attributes*

Attribute	Meaning
Title	Short recognizable description of the view
Stakeholders	List of stakeholders for whom the view is intended
Concerns	List of concerns that are addressed
Type of information	Free format description of the information that the view will contain
Presentation	The way in which the information will be presented, for instance as a book, as a report, or a slide-presentation, a help file, a website (with search engine?), etc.
Architecture concepts	*List of architecture concepts (from main concept map) to which the information is related. Used/needed attributes per concept*
Formal languages	*Which formal or informal language(s) will be used to describe the architecture?*
Modelling techniques	*Which formal or informal modeling technique(s) will be used to represent part of the architectural information?*
Analysis techniques	*Which analytical methods will be used to collect information that is needed?*
Outline of text	*Outline of the textual content of the view*
Stakeholder oriented terms to be used	*List of terms from the world of the intended stakeholders that will be used in the descriptions in this view*
List of diagrams	*List of diagrams that will be used*
Stakeholder oriented graphics to be used	*List of graphical images (icons) from the world of the intended stakeholders that will be used in the diagrams*

dom to decide which attributes of the template they wanted to use.

In the ING test case the IT architects chose for only the outline to delineate the view to be made, see Table 6. Together with the content-2-concerns table this viewpoint determines the view.

A goal of designing viewpoints is to present to each stakeholder exactly the information he needs regarding his concerns. When he is presented with more information, it should be easy for a stakeholder to select the information in which he is interested. This information should not be scattered throughout the documentation. On the architect side there is the limited time available and the practicalities of the distribution of the final documentation.

Table 6
Example of viewpoint, using only the 'outline' attribute

Target architecture Market Risks—Finance view

Problem statement (based on concerns of CIO and CEO Finance)
- How does the new proposed Market Risk Architecture relate to existing Finance Architecture?
- Textual summary of proposed Market Risk Architecture

Which problems will be solved by the new architecture?
- Concern 1 (one of more related columns from the content-2-concerns table for CIO or CEO Finance)
- Concern 2 (idem)
- Etc.

Which problems will be created by the new architecture?
- Dependency of Market Risk from Finance

Table 7
Example (short) viewpoint based on the template

Attribute	Content
Title	GUI view
Stakeholders	Judge, Registrar, Spokesman of police and justice, Representative of the lawyers
Concerns	• How to request the documents needed? • Registrar: How to update database?
Type of information	Visual impression of system. Possibilities of several (linked) screens
Presentation	Drawings on paper
Analysis techniques	Execute scenario(s) on drawing(s)
Stakeholder oriented terms	Search, update, screen, button, task, document, functionality
Stakeholder oriented graphics	Screen, buttons, tables, search fields, icons

For the extended attribute *Stakeholder oriented terms to be used* inspiration can be found in the list of concepts in the stakeholder profile. In the structure of the extended attribute *Outline of view* attention can be given to which concerns seem most pressing. Preferably the headings of the outline should be understandable for the stakeholder and serve by themselves as a summary of the architecture.

Table 7 shows an example of a (short) viewpoint, taken from one of the student cases.

Different modeling techniques can be combined in one view to represent all the information needed to address the concerns. See Clements et al. (2003) for a thorough description of proven modeling techniques. See Koning et al. (2002) for many practical guidelines concerning the design of diagrams.

4. Validation of the method

4.1. Overview

Over a period of eight months our method was discussed and tried out in various settings:

At the university one student did an individual assignment on software architecture in which he redesigned an existing IT-architecture document. The method was also part of a course on Software Architecture in which groups of students worked as an architecture team to design the architecture of a new system.

We asked for comments in the public mailing list 'IEEE 1471 group interested' and we also received some other comments from individuals. One of us was involved in teaching courses on software architecture to software engineers of a Dutch company. The stakeholder profiles and the condensed viewpoint template were used in the assignments.

With Ordina Public Consulting (OPC) the method was presented to and discussed by a group of 5 IT architects, two workshops were held in which the method was tried on generic stakeholder profiles and a generic architecture model (Jonkers et al., 2003), and a small group of consultants of Ordina has worked on redesigning an existing IT-architecture document.

The method was presented to two teams of two IT architects of ING who were working on a target architecture. It was decided that with our assistance the team which was closest to producing a final report would apply the method to their case. The other team would take part in the sessions.

Because of the experimental nature of the method, we chose not to try it out in projects with a high commercial risk. That does not mean that the test situations were not serious and that the participants were not dependent on the outcome. We encountered a serious attitude in all the test situations.

In the sessions we presented the method and answered questions, but we did not take part in the actual work of applying the method. Most of the work was done by the participants outside the meetings, without us being present. In the meeting we did not primarily pay attention to the deliverables, but focused on the attitudes of the participating architects and students. Were the steps clear to them? Were they motivated to perform the next step? Did they find the process meaningful?

We gathered 25 pages of notes from the test sessions.

4.2. Findings step 1: Stakeholder profiles

Step 1, compile stakeholder profiles, was generally received well by the participants in the sessions. They found it not difficult to do and it seemed to them an obvious step to begin with. We saw many good, clear profiles being made. A remark made was that this step should take into account IEEE Std 1471 clause 5.2 about stakeholders and concerns that should be identified at the minimum.

The main perceived immediate benefits were:

- It was helpful in imagining the stakeholders.
- It was a good way to concentrate on each stakeholder and get to his/her essentials.
- It made clear relevant differences between stakeholders.
- Within the design team it was helpful in coming to a shared perception of the role of a stakeholder.
- It is a fast way to present or check the perception with persons outside the team.

Some comments on the attributes of the template:

- Many participants preferred to formulate the goal of a stakeholder as a main 'task', instead of as a condition to be reached or maintained.
- Goals and tasks were helpful in finding concerns.
- Some doubted the usefulness of the 'concepts' attribute, but others found it helpful in picturing the stakeholder.
- Some participants added an extra attribute, a short free format textual description of the stakeholder.
- The concerns we saw sometimes were a rephrasing of the tasks, 'how can I... ', instead of real concerns.

4.3. Findings step 2: Summarize documentation

Step 2, summarizing the internal design documentation, was more of a challenge to the participants than step 1, the stakeholder profiles. It was more of an effort, but the result was rewarding. In the findings there is marked difference between the textual part and the graphical part.

First the textual part. For the textual summary we observed that it required an effort, and that there were doubts about it being possible, but after it was done the participants liked their own summary and it served well in step 3. We saw many, in our view, good summaries, that outlined in a few sentences the essentials of a new situation.

With the student groups we saw some statements in the textual summaries that were a rephrasing of a concern, 'the system will be user-friendly', instead of indications of the IT solution to provide this. We think this is partly because the summaries were made too early in the design process (requirements phase).

The main perceived immediate benefits of the textual summary were:

- It helped to articulate the essentials ('50 pages of information were reduced to three statements!').
- It forces to stay away from too much detailing.
- It is a good way to express design decisions and come to a common understanding within the team.

Then the graphical part. We pretty soon in some of the test sessions got negative feedback on the graphical part, the map of the main architectural concepts. It was considered difficult to produce and not useful. The relation to the textual summary was unclear.

In one case the concept map was perceived positive and in another there were concept maps already available. In these cases the perceived benefits were:

- It forces one to an even higher abstraction level than the textual summary.
- It gives a good overview of many aspects.

4.4. Findings step 3: Relate summary to concerns of stakeholders

Step 3 was less of a challenge than step 2. It required some explanation, but once that was given, it was more a matter of work, interesting work. We again make a difference between the textual part and the graphical part.

Filling in the content-2-concerns table of the textual part seemed to be the most meaningful activity for the participants. It stimulated rethinking the architecture design and aroused a higher level of involvement. Producing a more specific wording of a concern with regard to an architectural statement was found more difficult than adding details to an architectural statement with regard to a concern. Also the limitation of putting it all on one screen was felt by some.

The main perceived immediate benefits of relating the textual summary to the concerns of the stakeholders were:

- It gives a new insight in what the architecture design means to the stakeholders.
- It helps to stay focused on stakeholder concerns (instead of delving into technical details).
- It leads to strong, concise pieces of descriptive text.
- It is a good way to express further details of the design decisions and come to a common understanding.
- It is inspiring.

As said with the findings of step 2, the concept map of the graphical part gave some problems. Where it was made or already available and the concepts were related to the concerns of the stakeholders, two problems were reported: the diagrams get messy (this was amended by working in layers) and after a while you do not know anymore why you have connected a concern to a concept (this was amended later by adding text to the connecting lines). On the positive side, it was reported that the diagram easily showed the differences in the information needs of the stakeholders and that they lead to strong, concise pieces of descriptive text.

4.5. Findings step 4: Define viewpoints

Step 4, define viewpoints, was probably the step least appreciated. For this step we cannot offer a list of main perceived immediate benefits.

There was some disappointment about the fact that this step required more than expected analyzing/thinking/designing. Another point seemed to be the fundamental question of 'why make viewpoints? why not write the view right away?' Quite a few testers mixed view-description and view-content in the viewpoints. The terms 'view' and 'viewpoint' were found confusing by some. Where the views were produced, extra information was sometimes added without adjusting the viewpoints. The viewpoint template as described in Section 3.5 offered help, but also left many choices open.

Having said this, we can say we saw many viewpoints that, from first impression, outlined in a clear way the views to be made, which would address clearly stated concerns and which would be meaningful to the stakeholders.

We observed three analysis techniques used by the testers to make the transition from step 3 to step 4. Comparing the columns in the content-2-concerns table and combining similar columns in one view is one technique. Another approach is comparing the concerns that were used in step 3 to each other and creating groups of related concerns on gut feeling (but that does not make much use of the work of step 3). In one case we experimented with comparing the relations of concerns to concepts in the concept map.

One tester used the specifications of (sub)concerns in the Content-2-concerns table as headings in outlines of the views.

4.6. Findings concerning the whole method

From the test sessions we can report some miscellaneous feedback and observations that apply to the method as a whole.

We have received no reports on the method being used spontaneously in practice by the testers. One tester voiced his concern that application of the method in real life was too much work. Several participants in the sessions said that at the beginning they had some doubt about whether this method would really lead to results.

On various occasions the remark was made that communication is something that takes place during the whole design project and not only at the end. One participant suggested that the method should be applied several times during a design project on progressive versions of the design to ask comments from the stakeholders.

Because the method was presented early in the course the students applied it to requirements gathering and to the actual design work and not only for documenting results. A disadvantage of this is that the method is geared to converge to a small set of statements and which contradicts the investigative nature of the requirements phase. An advantage is that the students had a structuring mechanism that kept them on track.

On the whole the method forms a very straightforward process. During the process the already produced deliverables become a reference source. The steps in the method are clear and give structure to the working

sessions. The attention in the sessions is held until the end.

At the detail level the method needs more explanation. The attributes of the templates and the meaning of the cells in the table need to be described better.

We often had to urge for speed and for an intuitive, iterative approach. After the first learning stage, the time needed to apply this method diminishes greatly. The thinking process provoked by the method seemed to be of particular value to the architects.

5. Conclusions

5.1. Method 'viewpoint design' as a whole

From the findings in the previous section we conclude that the proposed method 'viewpoints design' is meaningful to practicing IT architects and leads to valuable results, but that improvements are necessary.

Application of the method is beneficial for expressing the positions of the various stakeholders, staying focused on the concerns of the stakeholders, reaching a higher abstraction level in the architectural design and for making clear what design information is needed to address which concerns.

Application of the method not only entails a roundup of existing design information, but brings with it some creative content production (more than we thought up front).

The techniques in the method can be applied not only at the end of the design project, but also during the project wherever design results need to be rationalized and communicated. We feel they are not suited to capture requirements, but they could very well be useful in design activities, wherever it comes to evaluating design alternatives against concerns of stakeholders.

The felt need to think about the way to communicate an architecture design is greater when: the architect has less experience, the architect is less familiar with the type of problem, there is no prescribed architecture framework.

5.2. Steps

The stakeholder profiles are not difficult to make and they function well to outline in a few words the position of a stakeholder.

The design summary is challenging, but gives a good grip on the design.

The content-2-concerns table of step 3 is a lot of work, but it is not difficult to make, it is a useful thinking tool and it is inspiring.

The concept map in steps 2 and 3 apparently is not suited for the average IT-architect busy in a design project, but it is probably more appropriate for people who have time and talent to reflect on architecture descriptions.

The viewpoint template works well to record the description of a view, but more support is needed for the transition from step 3 to step 4.

5.3. Improvements

The improvements most needed are: a better description of details of the method and more guidance in defining viewpoints. More guidance in defining viewpoints can consist of: guidelines for the transition from step 3 to step 4, example (library?) viewpoints, a more precise template, guidelines from other sources.

The concept map as the graphical design summary will not be made optional. We expect the concept map to give part of the needed extra support for the transition from step 3 to step 4.

5.4. IEEE 1471

Our findings indicate that the IEEE 1471 concepts 'stakeholder', 'concern' and 'view' are recognized and accepted by practicing IT-architects. The IEEE 1471 concept 'viewpoint' does not easily get operational significance. From the feedback received so far, we conclude that the average practicing IT architect in ongoing projects prefers to work from IEEE 1471 library viewpoints, instead of creating viewpoints from scratch on his own.

From the unexpected remaining gap between step 3 and step 4 we conclude that the orientation on stakeholders and their concerns, as prescribed by the IEEE Std 1471 conceptual model, is in itself insufficient to delineate architectural views. More factors have to be taken into account, like, for instance, the inherent relations between the key architectural concepts as expressed in the concept map. We expect that an in depth analysis of dimensions in existing architectural frameworks will reveal more criteria for structuring architectural information.

6. Future work

Primarily we intend to improve the method, see Section 5.3.

The improved method will be tried out in real life architecture projects.

It is our desire to perform an in depth analysis of dimensions in existing architectural frameworks, see Section 5.4.

In the background we would like to collect more practical experiences with applying IEEE Std 1471 to deepen our understanding of the standard.

Acknowledgments

We thank Rich Hilliard and the other reviewers for their valuable comments. We thank the participating companies for their support.

References

Baskerville, R., 1999. Investigating information systems with action research. Communications of the Association for Information Systems 2 (19), 1–31.

Boar, B.H., 1998. Constructing Blueprints for Enterprise IT Architectures. Wiley.

de Boer, F., Bonsangue, M., van Buuren, R., Groenewegen, L., Hoppenbrouwers, S., Lankhorst, M.M., Proper, H.A., Stam, A., van der Torre, L., Veldhuijzen van Zanten, G., 2003. In: Jonkers, H. (Ed.), Concepts for Architectural Description, Telematica Instituut, Enschede, The Netherlands. Available from: <http://archimate.telin.nl>.

Clements, P., Bachman, F., Bass, L., Garlan, D., Ivers, J., Little, R., Nord, R., Stafford, J., 2003. Documenting Software Architectures: Views and beyond. Addison-Wesley, Boston.

Gyselinck, V., Tardieu, H., 1998. The role of illustrations in text comprehension: What, when, for whom, and why. In: van Oostendorp, H., Goldman, S.R. (Eds.), The Construction of Mental Representations during Reading. Lea, ISBN 0-8058-2429-4.

Hilliard, R., 2001. Viewpoint modeling. In: Proceedings of 1st ICSE Workshop on Describing Software Architecture with UML.

IEEE, 2000. IEEE recommended practice for architecture description, IEEE, IEEE Standard 1471.

Koning, H., 2003. Method to enhance the stakeholder orientation while designing viewpoints, request for comments. Available from: <http://www.cs.vu.nl/~henk/index.php?page=2003>.

Koning, H., van Vliet, H., 2004. Real-life IT architecture design reports and their relation to IEEE Std 1471 stakeholders and concerns, submitted for publication.

Koning, H., Dormann, C., van Vliet, H., 2002. Practical guidelines for the readability of IT-architecture diagrams. In: Proceedings of the 20th Annual International Conference on Documentation, ACM SIGDOC 2002, Toronto, October 2002, pp. 90–99.

Kotonya, G., Sommerville, I., 1998. Requirements Engineering. John Wiley & Sons, Chichester, ISBN 0 471 97208 8.

Kruchten, P.B., 1995. The 4 + 1 view model of architecture. IEEE Software 12 (6), 42–50.

Parnas, D.L., Clements, P.C., 1986. A rational design process: How and why to fake it. IEEE Transactions on Software Engineering 12 (2), 251–257.

Soni, D., Nord, R.L., Hofmeister, C., 1995. Software architecture in industrial applications. In: Proceedings 17th International Conference on Software Engineering, ACM, pp. 196–207.

van Vliet, H., 2000. Software Engineering: Principles and Practice. John Wiley & Sons, Chichester.

van Welie, M., 2001. Task-based User Interface Design. Ph.D. Thesis. Vrije Universiteit, Amsterdam, The Netherlands. Available from: <http://www.cs.vu.nl/~martijn>.

Henk Koning has over 20 years practical experience in system development and architecture. This article is part of his Ph.D. study on "Communication of IT-architecture".

Hans van Vliet is a Professor in Software Engineering at the Vrije Universiteit. His research interests include software architecture and software measurement. Before joining the Vrije Universiteit, he worked as a researcher at the Centrum voor Wiskunde en Informatica (Amsterdam) and he spent a year as a visiting researcher at the IBM Almaden Research Center in San Jose, California. Hans has an M.Sc. in Computer Science from the Vrije Universiteit and a Ph.D. in Computer Science form the University of Amsterdam.